

ABUNDANCE EXTRACTIONS FROM AVIRIS IMAGE USING A SELF-ORGANIZING NEURAL NETWORK

D. Pedro Luis Aguilar,¹ D. Pablo Martínez Cobo,² D^a Rosa Pérez Utrero,³ D^a M^a Aránzazu Hormigo Nieves⁴

1.- INTRODUCTION.

Most of the luminous beams are formed by a mixture of radiation with a different wave length. The analysis of this luminous beam lets us obtain the intensity for every wavelength interval (channel) this shape is named hyperspectral signature.

The analysis of the signature from a sample lets us obtain remotely information about its constituents without destroying it. This possibility is not free from difficulties due to the amount of phenomena which intervene in the transmission and detection process what makes difficult the analysis of this information.

The AVIRIS sensor allows us to obtain the hyperspectral signature for every pixel of the image (and so for the corresponding element of the scene). The obtained hyperspectral image offers a large information about the scene with a considerable noisy component.

In the last few years many applications in this field have been presented (Kruse, 1990).

We describe the method proposed by Kruse consisting in some analysis steps:

- Minimum Noise Fraction (M.N.F)

This transformation was used to determine the inherent dimensionality of the AVIRIS data, similarly to the Principal Components transformations but maximizing the signal–noise ratio.

- Pixel Purity Index (P.P.I.)

The P.P.I. is a means of finding the most spectrally pure pixels which typically correspond to spectrally unique materials (endmembers). The P.P.I. is computed by repeatedly projecting n-dimensional scatterplots onto a random unit vector. The extreme pixel in each projection is marked and a counter (D.N.) indicates the total number of times they have been marked. In a P.P.I. image the pixel value is the D.N. counter.

- N-Dimensional Visualization

The P.P.I. image is used to select training sets by using this method.

- Unconstrained Linear Spectral Unmixing

To derive the apparent fractional abundance of each endmember material in each pixel, the endmember spectra matrix is inverted and multiplied by the observed spectra. (Kruse, 1990)

Other solutions based upon neural network have been presented (Aguilar, 1998), due to its parallelism and robustness with the noisy information.

¹ Computer Science Department, University of Extremadura, Spain. (paguilar@unex.es)

² Computer Science Department, University of Extremadura, Spain. (pablomar@unex.es)

³ Computer Science Department, University of Extremadura, Spain. (rosapere@unex.es)

⁴ Computer Science Department, University of Extremadura, Spain. (ahormigo@unex.es)

Our studies develop a self-organizing neural network for the analysis of hyperspectral images.

The self-organizing neural network (S.O.M.) is based on the competitive learning to form topologic maps. This way, after the learning process is finished, when presenting a spectrum to the network only the neurons closely related to the given spectrum are activated.

This kind of network was introduced by T. Kohonen (Kohonen, 1997) and has been successfully applied in many different fields: analysis and comprehension of images, artificial vision, optical characters reading, analysis and recognition of voice, musical and acoustic studies and telecommunications.

Within the spectroscopic analysis S.O.M. has been successfully applied in protein classification tasks, microbiology and infrared spectroscopy.

This work consists on studying the topologic maps formed by the self-organizing neural network to classify the pixels of a hyperspectral image in order to find the quantity of every element in a composite pixel (hyperspectral unmixing).

Most of the neural processing algorithms are computationally intensive and involve many iterative calculations. The formerly mentioned algorithms use basically matrix operations such as internal product and external product. In this way the MatLab© software pack offers an appropriate environment to carry out experiments with neural network, due to its facility for the matrix and vectorial notation and its graphic possibilities. This allows the fast implementation of neural computational models and the chance to test them.

2.- PROPOSED NEURAL MODEL.

A neuron is an information-processing unit that is fundamental to the operation of a neural network. We may describe three basic elements of the neuron model:

- A set of synapses or connecting links, characterized by its weight. Specifically, a signal x_j at the input of synapse j connected to neuron k is multiplied by the synaptic weight w_{kj} .
- An adder for summing the input signals, weighted by the respective synapses of the neuron.
- An activation function for limiting the amplitude of the output of a neuron.

A layered neural network is a set of neurons organized in the form of layers. In the simplest form of a layered network, we just have an input layer of source nodes that projects onto an output layer of neurons. The input layer has N units, and can be considered as a n -dimensional vector. The output layer has M units and can be also seen as a vector. Each unit owns N weights associated to the connections which come from the former layer, so the weight set is a matrix ($W_{M \times N}$). Fortunately, neural networks can often be described with matrixes. So the chosen tool to design the neural network must have vectorial and matrix facilities. (MatLab is our choice)

Our proposed network architecture has two sub-nets (Aguilar, 1998):

The first one is a one-dimensional S.O.M. with N input neurons (one for each channel) and M output neurons (one for each endmember).

The second sub-net is a Hopfield-like neural network with M neurons fully interconnected. Figure 1 depicts this architecture.

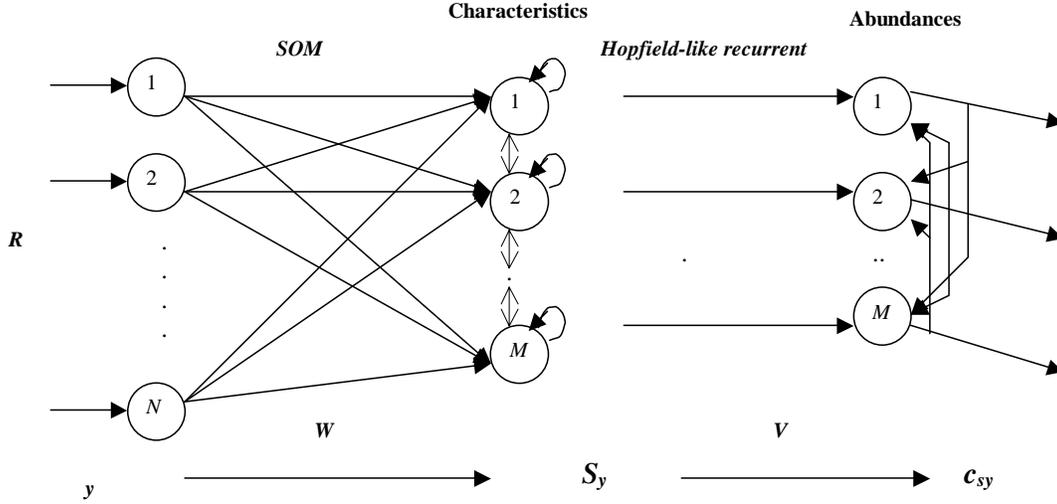


Figure 1. Proposed architecture.

2.1 Self-organizing layer

We consider the one-dimensional lattice of M neurons, which contains two different types of connections. There are feedforward connections from the N input layer neurons, and those that are internal to the network by virtue of self-feedback and lateral feedback. These two types of local connections serve two different purposes. The weighted sum of the input signals at each neuron is designed to perform feature detection. Hence each neuron produces a selective response to a particular set of input signals. The feedback connections, on the other hand, produce excitatory or inhibitory effects, depending on the distance from the neuron (Kohonen, 1989).

There are five basic steps involved in the application of the training algorithm. These five steps are repeated until the map formation is completed. The algorithm is summarized as follows:

1. Initialization

Choose random values for the initial weight vectors $w_i(0)$ ($i = 1, 2, \dots, M$). It is desirable to keep the magnitude of the weights small.

2. Sampling

Choose an input pattern $x(n)$ belonging to the learning patterns (or references R). The selection can be randomly.

3. Similarity Matching

Find the best-matching (winning) neuron i^* at the time t , using the minimum-distance Euclidean criterion:

$$i^*[x(n)] = \min_j \|x(n) - w_j(t)\| \quad j = 1, 2, \dots, M \quad (1)$$

Updating

Adjust the synaptic weight vectors of all neurons, using the update formula:

$$w_i(t+1) = w_i(t) + \eta(t) \sigma(t, i, i^*[x(n)]) (x(n) - w_i(t)) \quad (2)$$

where $\eta(t)$ is the learning-rate parameter, and $\sigma(t, i, i^*[x(n)])$ is the neighborhood function centered around the winning neuron.

4. Continuation

Continue with step 2 until no noticeable changes in the weights space are observed, or the maximum time is been achieved

2.2 The second layer

The second layer is a recurrent network (i.e. a neural network with feedback) made up of M identical neurons.

This network is used for determining the components in a composite spectrum as function of the Self-organizing network output when the mixture spectrum has been considered as input (Perez,1996)(Perez,1999).

To solve this problem by adaptive methods we propose an optimization process. We define the following quadratic function

$$\phi(\mathbf{c}) = \frac{1}{2} \mathbf{c}^T \mathbf{S}_w \mathbf{c} - \mathbf{c}^T \mathbf{s}_y \quad (3)$$

where \mathbf{c} is the contribution vector, \mathbf{S}_w is the topology mapping obtained from the first network to the prototypes \mathbf{W} and \mathbf{s}_y is the same transformation to the composite spectrum \mathbf{y} .

Our objective is to find the vector \mathbf{c} that minimizes ϕ . For this purpose we use an iterative process based on the feedback network characteristics whose iterations will converge to a solution that minimizes a network energy function defined as (Hopfield, 1986):

$$E = -\eta \frac{1}{2} \mathbf{x}^T \mathbf{V} \mathbf{x} + \eta \theta \mathbf{x} \quad (4)$$

where η is the learning parameter chosen as $1/\text{Trace}(\mathbf{V})$, \mathbf{V} is the weight matrix, θ is the threshold vector, and \mathbf{x} is the net values.

The iterative process can be defined as:

$$\Delta \mathbf{c}(t) = -\eta \mathbf{S}_w \mathbf{c}(t) + \eta \mathbf{s}_y \quad (5)$$

Then, the operation network is described as:

$$\mathbf{c}(t+1) = \mathbf{c}(t) - \eta \mathbf{S}_w \mathbf{c}(t) + \eta \mathbf{s}_y = [\mathbf{I} - \eta \mathbf{S}_w] \mathbf{c}(t) + \eta \mathbf{s}_y \quad (6)$$

Where \mathbf{I} is the identity matrix.

3.-LEARNING.

There are two phases to the operation of the neural model proposed in the second layer:

- * The storage phase.
- * The iterative process.

In the first one, according to the inner product rule, the synaptic weights get defined. And during the iterative process, each neuron updates its state until there are no further changes to report.

The first layer of the neural network must be trained with spectral signatures from the pixels of the image. This learning process is very important in the neural network performance (noise cancellation and endmember spectra extraction).

The number of endmember abundance to obtain, determines the number of neurons on the second layer of the SOM.

The selection of the training subset and the order to learn each pixel spectra must be chosen to optimize the endmember selection.

We have evaluated two possibilities:

- a) Use the whole spectral signatures set of the image, sequentially from the pixel (0,0) to the last pixel. This method is computationally intensive and takes a long time to be finished. The endmember learnt by the S.O.M. might include spectral measures from all the pixels of the image.
- b) Use a subset of spectral signatures for each endmember to be learnt. The subset for each endmember can be selected graphically on the image using the knowledge the user has on the scene.

After the training process, it is easy to compare the spectral signature of each endmember to the spectral signature of some pixels of the subset to identify the endmembers.

4.-RESULTS

The proposed methodology has been widely checked especially in endmember extraction phase.

In this way figure 2 shows the endmember hyperspectral signature from figure 4 extracted by a S.O.M. with 3 output neurons using the whole image. The hyperspectral signature of the corresponding pixel of the image is shown in figure 3.

The noise reduction achieved by the S.O.M. in this process can be seen by comparing figure 2 and figure 3 denoting a robust process.

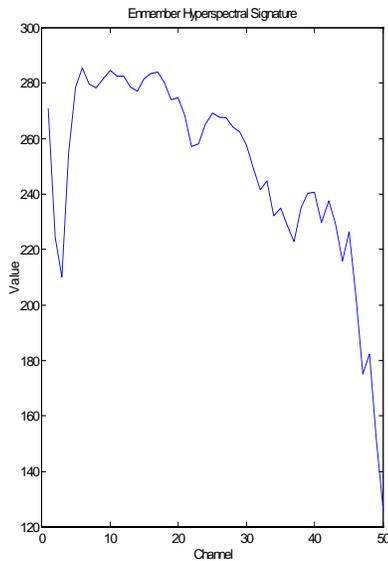


Figure 2.-Endmember hyperspectral signature

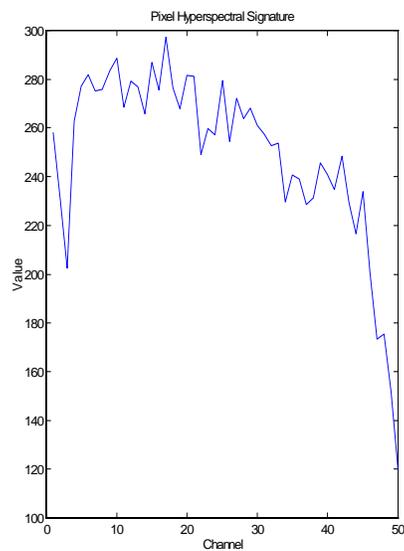


Figure 3.-Pixel hyperspectral signature

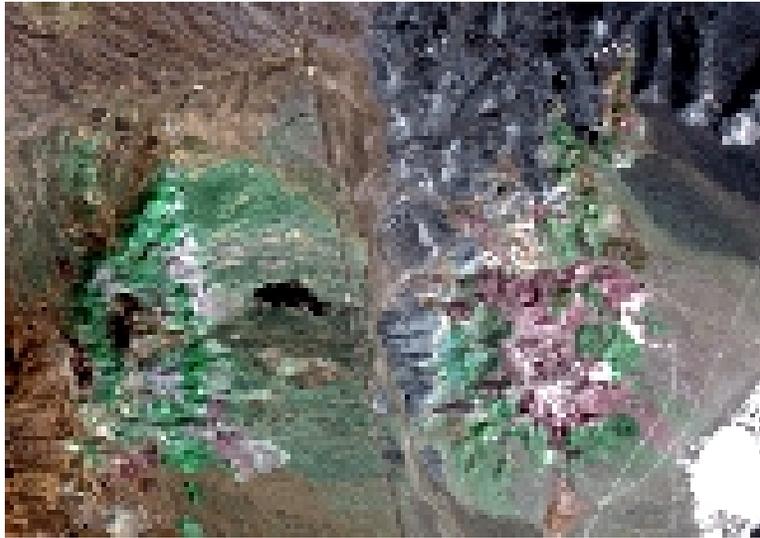


Figure 4.- AVIRIS image

The robustness of the proposed algorithm was evaluated corrupting the hyperspectral signatures of the image with different noise ratio, the proposed network builds appropriately the endmember even for high noise ratio percentage.

The influence of the noise ratio on the relative error can be seen in figure 5 showing a good performance independently of the ratio.

Many pixels on the image are closely related, one of the goals of an unmixing algorithm is its capability to resolve similar hyperspectral signatures.

In order to measure the endmembers similarity we use the spectra correlation index .

Figure 6 shows the relative error obtained at evaluating the abundance versus the correlation index. The algorithm presents a good performance with the correlation index.

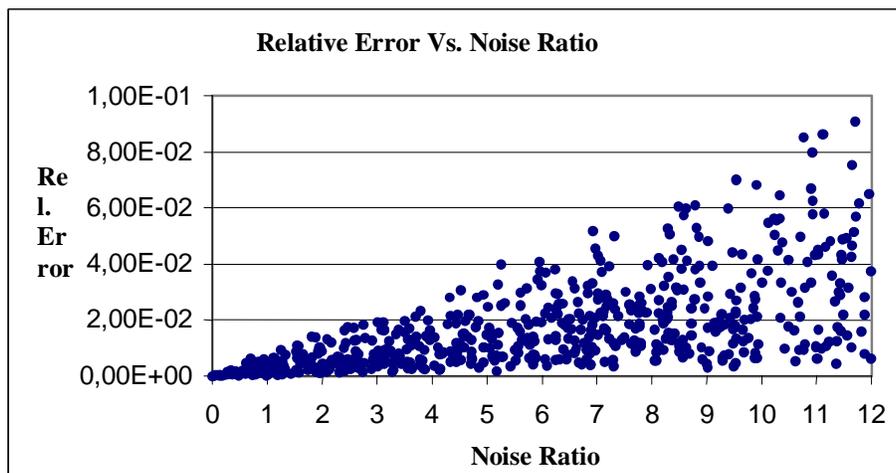


Figure 5.- Influence of noise ratio

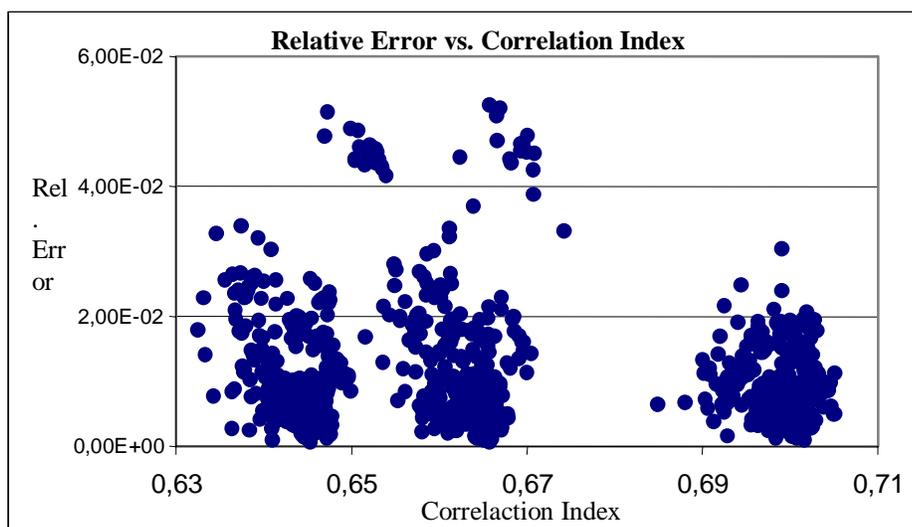


Figure 6.- Influence of the correlation Index

5.- CONCLUSIONS

We have improved an AVIRIS neural tool including an S.O.M. layer.

The new tool resolves adequately hyperspectral images to make easier the endmember extraction.

The algorithm is robust with the noise-ratio percentage ratio and it presents a good performance with the correlation index.

REFERENCES

- Aguilar, P.L. & P. Bachiller & P. Martínez & A. Merchán & R.M. Pérez, February 1, 1998, "Spectra Evaluation and Recognition in the Mixture Problem Using SOFM Algorithm," Proc. of the International Symposium on Engineering of Intelligent Systems, (*EIS'98*), Vol. 2 , pp. 118-124.
- Holpfield, J.J., 1982, "Neural Networks and Physical Systems with Emergent Collective Computational Abilities," Proc. of National Academy of Sciences, 79, pp. 2554-2558.
- Kohonen, T. , 1995-1997, "Self-Organization Maps (2nd. ed.)," Springer Series in Information Science, ISBN 3-540-62017-6.
- Kruse, .F.A., 1996, "Geologic Mapping Using Combined Analysis of Airbone Visible/Infrared Imaging Spectrometer (AVIRIS) and SIR-C/X-SAR Data." In Proceedings, International Symposium on Optical Science, Engineering and Instrumentation, SPIE Proceedings, August 1996, Denver, v. 2819, pp. 24- 35.
- Pérez , R.M. & P.L. Aguilar & Silva A & P. Martínez, November 1996, "Adaptive algorithm to solve the mixture problem with neural network methodology," 3rd International Workshop in Signal/Processing, Manchester.
- Pérez , R.M. & P.L. Aguilar & P. Bachiller & P. Martínez, 1999, "Neural network quantifier for solving the mixture problem and its implementation by systolic array," *Microelectronics Journal*, pp. 77-82.